III Semester

Course 5: Object Oriented Programming using Java

Credits -3

Course Objectives

To introduce the fundamental concepts of Object-Oriented programming and to design &implement object-oriented programming concepts in Java.

Course Outcomes

Upon successful completion of the course, a student will be able to:

1. Understand the basic concepts of Object-Oriented Programming and Java Program Constructs
2. Implement classes and objects and analyze Inheritance and Dynamic Method Dispatch
3. Demonstrate various classes in different packages and can design own packages
4. Manage Exceptions and Apply Threads
5. Create GUI screens along with event handling

UNIT-I

OOPs Concepts and Java Programming: Introduction to Object-Oriented concepts, procedural and object-oriented programming paradigm

Java programming: An Overview of Java, Java Environment, Data types, Variables, constants, scope and life time of variables, operators, type conversion and casting, Accepting Input from the Keyboard, Reading Input with Java.util.Scanner Class, Displaying Output with System.out.printf(), Displaying Formatted Output with String.format(), Control Statements

UNIT-II

Arrays, Command Line Arguments, Strings-String Class Methods

Classes & Objects: Creating Classes, declaring objects, Methods, parameter passing, static fields and methods, Constructors, and 'this' keyword, overloading methods and access

Inheritance: Inheritance hierarchies, super and subclasses, member access rules, 'super' keyword, preventing inheritance: final classes and methods, the object class and its methods; **Polymorphism**: Dynamic binding, method overriding, abstract classes and methods;

UNIT-III

Interface: Interfaces VS Abstract classes, defining an interface, implement interfaces, accessing implementations through interface references, extending interface;

Packages: Defining, creating and accessing a package, understanding CLASSPATH, importing packages.

Exception Handling: Benefits of exception handling, the classification of exceptions, exception hierarchy, checked exceptions and unchecked exceptions, usage of try, catch, throw, throws and finally, rethrowing exceptions, exception specification, built in exceptions, creating own exception sub classes.

UNIT-IV

Multithreading: Differences between multiple processes and multiple threads, thread states, thread life cycle, creating threads, interrupting threads, thread priorities, synchronizing threads, inter thread communication.

Stream based I/O (java.io) – The Stream classes-Byte streams and Character streams, Reading console Input and Writing Console Output, File class, Reading and writing Files, The Console class, Serialization

UNIT-V

GUI Programming with Swing- Introduction, MVC architecture, components, containers. Understanding Layout Managers - Flow Layout, Border Layout, Grid Layout, Card Layout, GridBag Layout.

Event Handling- The Delegation event model- Events, Event sources, Event Listeners, Event classes, Handling mouse and keyboard events, Adapter classes, Inner classes, Anonymous Inner classes.

Text Books:

1. Java The complete reference, 9th edition, Herbert Schildt, McGraw Hill.
2. Understanding Object-Oriented Programming with Java, updated edition, T. Budd, Pearson Education.

Reference Books

1. Cay S. Horstmann, "Core Java Fundamentals", Volume 1, 11 th Edition, Prentice Hall, 2018.
2. Paul Deitel, Harvey Deitel, "Java SE 8 for programmers", 3rd Edition, Pearson, 2015.
3. S. Malhotra, S. Chudhary, Programming in Java, 2nd edition, Oxford Univ. Press.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Quiz on Object-Oriented Programming Concepts and Java Constructs

Evaluation Method: Quiz Performance and Knowledge Retention

Unit 2: Activity: Object-Oriented Programming Assignment: Class Implementation

Evaluation Method: Assignment Completion and Correctness

Unit 3: Activity: Hands-on Lab Activity: Creating and Using Custom Java Packages

Evaluation Method: Lab Performance and Correctness of Code Implementation

Unit 4: Activity: Case Study Discussion on where multi-threading is crucial

Evaluation Method: Critical thinking, problem-solving, and presentation skills.

Unit 5: Activity: GUI design contest using Java Swings

Evaluation Method: GUI design, Visual appearance and user friendliness, usability, and adherence to event handling principles.

III Semester

Course 5: Object Oriented Programming using Java Lab

Credits -1

List of Experiments

1. Write a Java program to print Fibonacci series using for loop.
2. Write a Java program to calculate multiplication of 2 matrices.
3. Create a class Rectangle. The class has attributes length and width. It should have methods that calculate the perimeter and area of the rectangle. It should have read Attributes method to read length and width from user.
4. Write a Java program that implements method overloading.
5. Write a Java program for sorting a given list of names in ascending order.
6. Write a Java program that displays the number of characters, lines and words in a text file.
7. Write a Java program to implement various types of inheritance

   i. Single ii. Multi-Level iii. Hierarchical iv. Hybrid
8. Write a java program to implement runtime polymorphism.
9. Write a Java program which accepts withdraw amount from the user and throws an exception "In Sufficient Funds" when withdraw amount more than available amount.
10. Write a Java program to create three threads and that displays "good morning", for every one second, "hello" for every 2 seconds and "welcome" for every 3 seconds by using extending Thread class.
11. Write a Java program that creates three threads. First thread displays "OOPS", the second thread displays "Through" and the third thread Displays "JAVA" by using Runnable interface.
12. Implement a Java program for handling mouse events when the mouse entered, exited, clicked, pressed, released, dragged and moved in the client area.
13. Implement a Java program for handling key events when the key board is pressed, released, typed.
14. Write a Java swing program that reads two numbers from two separate text fields and display sum of two numbers in third text field when button "add" is pressed.
15. Write a Java program to design student registration form using Swing Controls. The form which having the following fields and button SAVE

    Form Fields are: Name, RNO, Mailid, Gender, Branch, Address.

III Semester

Course 6: Data Structures using C

Credits -3

Course Objectives

To introduce the fundamental concept of data structures and to emphasize the importance of various data structures in developing and implementing efficient algorithms.

Course Outcomes

Upon successful completion of the course, a student will be able to:

1. Understand various Data Structures for data storage and processing.
2. Realize Linked List Data Structure for various operations
3. Analyze step by step and develop algorithms to solve real world problems by implementing Stacks, Queues data structures.
4. Understand and implement various searching & sorting techniques.
5. Understand the Non-Linear Data Structures such as Binary Trees and Graphs

UNIT-I

Basic Concepts: Pointers and dynamic memory allocation, Algorithm-Definition and characteristics, Algorithm Analysis-Space Complexity, Time Complexity, Asymptotic Notation **Introduction to Data structures:** Definition**,** Types of Data structure, Abstract Data Types (ADT), Difference between Abstract Data Types, Data Types, and Data Structures.

Arrays-Concept of Arrays, Single dimensional array, Two dimensional array, Operations on arrays with Algorithms (searching, traversing, inserting, deleting)

UNIT-II

Linked List: Concept of Linked Lists, Representation of linked lists in Memory, Comparison between Linked List and Array, Types of Linked Lists - Singly Linked list, Doubly Linked list, Circularly Singly Linked list, Circularly Doubly Linked list;

Implementation of Linked List ADT: Creating a List, Traversing a linked list, Searching linkedlist, Insertion and deletion into linked list (At first Node, Specified Position, Last node), Application of linked lists

UNIT-III

Stacks: Introduction to stack ADT, Representation of stacks with array and Linked List, Implementation of stacks, Application of stacks - Polish Notations - Converting Infix to Post Fix Notation - Evaluation of Post Fix Notation - Tower of Hanoi, Recursion: Concept and Comparison between recursion and Iteration

Queues: Introduction to Queue ADT, Representation of Queues with array and Linked List, Implementation of Queues, Application of Queues Types of Queues- Circular Queues, De-queues, Priority Queue

UNIT-IV

Searching: Linear or Sequential Search, Binary Search and Indexed Sequential Search

Sorting: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort and Merge Sort

UNIT-V

Binary Trees: Concept of Non- Linear Data Structures, Introduction Binary Trees, Types of Trees, Basic Definition of Binary Trees, Properties of Binary Trees, Representation of Binary Trees, Operations on a Binary Search Tree, Binary Tree Traversal, Applications of Binary Tree.

Graphs: Introduction to Graphs, Terms Associated with Graphs, Sequential Representation of Graphs, Linked Representation of Graphs, Traversal of Graphs (DFS, BFS), Application of Graphs.

Text Books:

1. Horowitz and Sahani, "Fundamentals of Data Structures", Galgotia Publications Pvt Ltd Delhi India.
2. A.K. Sharma ,Data Structure Using C, Pearson Education India.
3. "Data Structures Using C" Balagurusamy E. TMH

Reference Books

1. "Data Structures through C", Yashavant Kanetkar, BPB Publications
2. Rajesh K. Shukla, "Data Structure Using C and C++" Wiley Dreamtech Publication.
3. Lipschutz, "Data Structures" Schaum's Outline Series, Tata Mcgraw-hill Education (India)Pvt. Ltd .
4. Michael T. Goodrich, Roberto Tamassia, David M. Mount "Data Structures and Algorithms in C++", Wiley India.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Algorithm analysis exercises

Evaluation Method: Programming Assignment and Correctness

Unit 2: Activity: Presentations on real-life applications of linked lists

Evaluation Method: Presentation skills or reports

Unit 3: Activity: Role-playing activities for stack operations

Evaluation Method: Problem-solving skills, communication and collaboration abilities.

Unit 4: Activity: Sorting algorithm analysis and comparison activities

Evaluation Method: Performance analysis and presentation.

Unit 5: Activity: Case Study on Applications of Graphs

Evaluation Method: Critical thinking, problem-solving, and presentation skills

III Semester

Course 6: Data Structures Using C

Credits -1

List of Experiments:

1. Write a program to read 'N' numbers of elements into an array and also perform the following operation on an array
    a. Add an element at the beginning of an array
    b. Insert an element at given index of array
    c. Update an element using a values and index
    d. Delete an existing element
2. Write Program to implement Single Linked List with insertion, deletion and traversal operations
3. Write Program to implement Circular doubly Linked List with insertion, deletion and traversal operations
4. Write Programs to implement the Stack operations using an array
5. Write a program using stacks to convert a given infix expression to postfix
6. Write Programs to implement the Stack operations using Liked List.
7. Write Programs to implement the Queue operations using an array.
8. Write Programs to implement the Queue operations using Liked List.
9. Write a program for Binary Search Tree Traversals
10. Write a program to search an item in a given list using the following Searching Algorithms
    a. Linear Search
    b. Binary Search.
11. Write a program for implementation of the following Sorting Algorithms
    a. Bubble Sort
    b. Insertion Sort
    c. Quick Sort

III Semester

Course 7: Computer Organization

Credits -3

Course Objectives

To familiarize with organizational aspects of memory, processor and I/O.

Course Outcomes

Upon successful completion of the course, the students will be able to

1. Identify different types of instructions
2. Differentiate between micro-programmed and hard-wired control units.
3. Analyse the performance of hierarchical organization of memory.
4. Summarize different data transfer techniques.
5. Demonstrate arithmetic operations on fixed- and floating-point numbers and illustrate concepts of parallel processing.

UNIT – I

Register Transfer Language and Micro Operations: Introduction- Functional units, computer registers, register transfer language, register transfer, bus and memory transfers, arithmetic, logic and shift micro-operations, arithmetic logic shift unit.

Basic Computer Organization and Design: Instruction codes, instruction cycle.

Register reference instructions, Memory – reference instructions, input – output and interrupt.

UNIT – II

CPU and Micro Programmed Control: Central Processing unit: Introduction, instruction formats, addressing modes.Control memory, address sequencing, design of control unit - hard wired control, micro programmed control.

UNIT – III

Memory Organization: Memory hierarchy, main memory, auxiliary memory, associative memory, cache Memory and mappings.

UNIT – IV

Input-Output Organization: Peripheral Devices, input-output interface, asynchronous datatransfer, modes of transfer- programmed I/O, priority interrupt, direct memory access, Input – Output Processor (IOP).

UNIT – V

Computer Arithmetic and Parallel Processing: Data representation- fixed point, floating point, addition and subtraction, multiplication and division algorithms.

Parallel Processing-Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline.

Text Books:

1. M. Moris Mano, “Computer Systems Architecture”, 3rd edition, Pearson/ PHI.

Reference Books:

1. Carl Hamacher, ZvonksVranesic, SafeaZaky, “Computer Organization”, 5th edition,McGraw Hill.
2. William Stallings, “Computer Organization and Architecture”, 8th edition, Pearson/PHI.

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Quiz competition on micro-operations.
Evaluation Method: Accuracy and speed in answering quiz questions.

Unit 2: Activity: Instruction Format Puzzle: Solving a puzzle to decode and understand instruction formats.
Evaluation Method: Accuracy and speed in completing the puzzle.

Unit 3: Activity: Memory Hierarchy Poster: Creating informative posters or infographics onmemory hierarchy.
Evaluation Method: Clarity of information, presentation and creativity of visual design.

Unit 4: Activity: I/O Troubleshooting Challenge
Evaluation Method: problem identification, feasibility of proposed solutions, and clarity of explanations.

Unit 5: Activity: Case Study on Parallel processing architecture.
Evaluation Method: Understanding of parallel processing concepts and architectures.

III Semester

Course 3: Computer Organization

Credits -1

Lab Experiments

1. Implement a C program to convert a Hexadecimal, octal, and binary number to decimal number vice versa.
2. Implement a C program to perform Binary Addition & Subtraction.
3. Implement a C program to perform Multiplication of two binary numbers.
4. Implement arithmetic micro-operations using logic gates.
5. Implement logic and shift micro-operations using logic gates.
6. Implement a C program to perform Multiplication of two binary numbers (signed) using Booth’s Algorithms.
7. Implement a C program to perform division of two binary numbers (Unsigned) using restoring division algorithm.
8. Implement a C program to perform division of two binary numbers (Unsigned) using non-restoring division algorithm.
9. Write assembly language code for A+B*(C-D) using various instruction formats in MASMor any open-source assembler.
10. Write assembly language code for A+B*C using various addressing modes in MASM or any open-source assembler.

III Semester

Course 8: Operating Systems

Credits -3

Course Objectives

To gain knowledge about various functions of an operating system like memory management, process management, device management, etc.

Course Outcomes:

Upon successful completion of the course, a student will be able to:

1. Demonstrate knowledge and comprehension of operating system functions.
2. Analyze different process scheduling algorithms and apply them to manage processes and threads effectively
3. Create strategies to prevent, detect, and recover from deadlocks, and design solutions for inter-process communication and synchronization problems.
4. Compare and contrast different memory allocation strategies and evaluate their effectiveness
5. Evaluate disk scheduling algorithms while implementing OS security measures

UNIT- I

What is Operating System? History and Evolution of OS, Basic OS functions, Resource Abstraction, Types of Operating Systems– Multiprogramming Systems, Batch Systems, Time Sharing Systems; Operating Systems for Personal Computers, Workstations and Hand-held Devices, Process Control & Real time Systems.

UNIT- II

Processor and User Modes, Kernels, System Calls and System Programs, System View of the Process and Resources, Process Abstraction, Process Hierarchy, Threads, Threading Issues, Thread Libraries; Process Scheduling- Non-Preemptive and Preemptive Scheduling Algorithms.

UNIT III

Process Management: Deadlock, Deadlock Characterization, Necessary and Sufficient Conditions for Deadlock, Deadlock Handling Approaches: Deadlock Prevention, Deadlock Avoidance and Deadlock Detection and Recovery.

Concurrent and Dependent Processes, Critical Section, Semaphores, Methods for Inter process Communication; Process Synchronization, Classical Process Synchronization Problems: Producer-Consumer, Reader-Writer.

UNIT IV

Memory Management: Physical and Virtual Address Space; Memory Allocation Strategies–Fixed and -Variable Partitions, Paging, Segmentation, Virtual Memory.

UNIT V

File and I/O Management, OS security: Directory Structure, File Operations, File Allocation Methods, Device Management, Pipes, Buffer, Shared Memory, Disk Scheduling algorithms.

Text Books:

1. Operating System Principles by Abraham Silberschatz, Peter Baer Galvin and GregGagne (7th Edition) Wiley India Edition.

Reference Books

1. Operating Systems: Internals and Design Principles by Stallings (Pearson)
2. Operating Systems by J. Archer Harris (Author), Jyoti Singh (Author) (TMH)

SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:

Unit 1: Activity: Case Study on a specific Operating System: highlighting its functions and key features.

Evaluation Method: Case study presentation, depth of understanding of operating system functions, and ability to articulate key concepts.

Unit 2: Activity: Comparison Poster on Scheduling Algorithms

Evaluation Method: Assessment of posters based on content accuracy, clarity of information, visual presentation, and ability to convey key insights.

Unit 3: Activity: Assignment on Dead Lock prevention techniques

Evaluation Method: Understanding, Completion and report.

Unit 4: Activity: Debate on various Memory allocation schemes

Evaluation Method: Debate arguments, ability to counter opposing viewpoints, logical reasoning, and presentation skills.

Unit 5: Activity: Comparative study of various disk scheduling algorithms using real world datasets

Evaluation Method: Analysis methodology, accuracy of results, and presentation of findings and conclusions.

III Semester

Course 8: Operating Systems

Credits -1

List of Experiments:

1. Illustrate the LINUX commands
    a) pwd
    b) mkdir
    c) rmdir
    d) grep
    e) chmod
    f) ls
    g) rm
    h) cp
2. Write a program to calculate average waiting time and turn around time of each process using the following CPU Scheduling algorithm for the given process schedules.
    a) FCFS
    b) SJF
    c) Priority
    d) Round Robin
3. Simulate MVT and MFT memory management techniques
4. Write a program for Bankers Algorithm for Dead Lock Avoidance
5. Implement Bankers Algorithm Dead Lock Prevention.
6. Write a program to simulate Producer-Consumer problem.
7. Simulate all Page replacement algorithms.
    e) FIFO
    f) LRU
    g) LFU
    h) Optimal
8. Simulate Paging Techniques of memory management
9. Simulate the following disk scheduling algorithms
    a) FCFS
    b) SSTF
    c) SCAN
    d) CSCAN